SINERGI Vol. 29, No. 2, June 2025: 485-496
http://publikasi.mercubuana.ac.id/index.php/sinergi
http://doi.org/10.22441/sinergi.2025.5.018

Comparative study of CNN techniques for tuberculosis detection using chest X-ray images from Indonesia

Suci Dwijayanti*, Regan Agam, Bhakti Yudho Suprapto
Department of Electrical Engineering, Faculty of Engineering, Universitas Sriwijaya, Indonesia

Abstract

Convolutional neural networks (CNNs) represent a popular deep-learning approach for image classification tasks. They have been extensively employed in studies aimed at classifying tuberculosis (TB), coronavirus disease 2019 (COVID-19), and normal conditions on chest X-ray images. However, there is limited research utilizing Indonesian data, and the integration of CNN models into user-friendly interfaces accessible to healthcare professionals remains uncommon. This study addresses these gaps by employing three CNN architectures—AlexNet, LeNet, and a modified model—to classify TB, COVID-19, and normal condition images. Training data were sourced from both a local hospital in Indonesia (RSUP dr. Rivai Abdullah) and an additional online dataset. Results indicate that AlexNet achieved the highest accuracy, with rates of 97.52%, 64.45%, and 92.43% on the Kaggle dataset, the RSUP Dr. Rivai Abdullah dataset, and the combined dataset, respectively. Subsequently, this model was integrated into a user interface and deployed for testing using new data from the RSUP Dr. Rivai Abdullah dataset. The web-based interface, powered by the Gradio library, successfully detected 7 out of 10 new cases with 70% accuracy. This implementation may enable medical professionals to make preliminary diagnoses.

Keywords:
AlexNet;
Deployment;
Indonesian Dataset;
LeNet;
Modified architecture;

Article History:
Received: October 17, 2024
Revised: January 6, 2025
Accepted: February 4, 2025
Published: May 15, 2025

Corresponding Author:
Suci Dwijayanti
Department of Electrical Engineering, Faculty of Engineering, Universitas Sriwijaya, Indonesia
Email:
sucidwijayanti@ft.unsri.ac.id

INTRODUCTION

Tuberculosis is a contagious disease that usually affects the lungs and is caused by a bacterial infection, namely, *Mycobacterium tuberculosis* (TB) [1]. The identification and diagnosis of TB are typically performed using chest X-ray images of the lungs. These images are examined by specialist radiologists to determine the final patient diagnosis. However, the number of specialist radiologists and the daily volume of TB patients undergoing X-ray imaging are often disproportionate, particularly in developing countries such as Indonesia. As a result, healthcare workers or radiologists may need more time to analyze each chest X-ray image, potentially leading to identification and diagnosis errors due to fatigue caused by inadequate rest. In Indonesia, which ranks second after India in terms of TB cases, 969,000 cases and 93,000 deaths are recorded annually, equivalent to 11 deaths per hour [2].

Efforts have been made to assist professionals in identifying TB using machine learning (ML) with chest X-ray imaging data, as observed in [3, 4, 5], where research was conducted to identify TB using convolutional neural network (CNN) models with datasets from China. Moreover, [6] utilized the VGG-16 architecture to detect tuberculosis using X-ray image data from the NIAID TB dataset, and [7] compared nine different CNN models to classify TB. Additionally, [8] modified a method to improve the quality of TB chest X-rays using deep learning and the Shenzhen dataset.

An ML model was then developed using an artificial neural network [9][10]. Subsequently, [11] utilized K-Nearest Neighbors (KNN) with HOG feature extraction. Meanwhile, [12] proposed multitask optimization using a genetic algorithm to select the features to be fed into the support vector machine. However, these methods rely on features extracted from the X-ray images. To address this, deep learning approaches have been implemented, as noted in [13, 14, 15]. Nevertheless, these studies primarily use global datasets. Furthermore, they were limited to creating and optimizing ML models and did not involve deployment processes, such as developing a user interface (UI) for healthcare professionals or radiology experts. In studies that did deploy ML models, such as [16][17], python libraries and frameworks, including Bottle and Gradio, were used. However, research focusing on deploying ML models in UIs is still limited, especially for TB detection. Moreover, these studies utilized publicly accessible datasets from Kaggle [6][18] and datasets from China [3, 4, 5, 16]. Data from Indonesia are limited and have only been presented in a few studies [19].

To address this issue this study implements CNN methods with a private dataset containing chest X-ray images obtained from the Palembang city region in Indonesia. Unlike [20][21], which compared various traditional machine learning methods using the Brazil dataset, in this study, we compared three CNN architectures for identifying TB in the Palembang region of Indonesia. Subsequently, the CNN model is deployed in the form of a UI using the Gradio library. This library offers the advantage of enabling the creation of UIs for previously developed ML models, making them easier to access, such as by sharing URLs and applications. Furthermore, the resulting web interface is simpler and user-friendly. The deployment of this ML model is expected to assist doctors in rapidly and accurately identifying and diagnosing chest X-ray images automatically.

The contributions of this study are as follows:

(1) We developed a user-friendly website interface for identifying TB through X-ray images using a CNN with an Indonesian dataset,

(2) We compared three architectures of deep learning for identifying TB.

This paper is organized as follows. Section 1 provides the background of the research. Section 2 describes the methods used. The results and discussion are presented in Section 3. Finally, the conclusions are presented in Section 4.

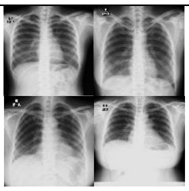
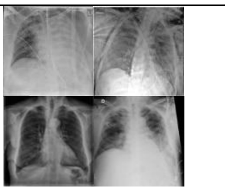
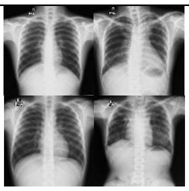
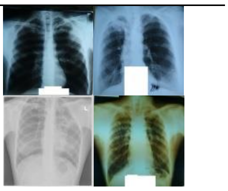
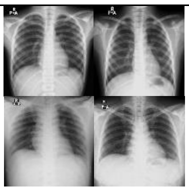
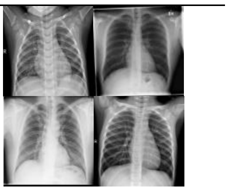
METHOD

The data used in this research were sourced from the local hospital RSUP Dr. Rivai Abdullah in the form of chest X-ray images representing TB patients in the South Sumatra region. Additionally, data from Kaggle were utilized [12, 13, 14, 15, 16]. The collected data include chest X-ray images from TB, normal, and COVID-19 cases, with a total of 141 normal, 149 TB, and 279 COVID-19 data. These data were obtained from residents of Banyuasin and its surrounding areas. Apart from the data acquired from RSUP Ddr. Rivai Abdullah, other data were obtained from Kaggle [7].

The Kaggle TB and normal case dataset consists of a combination of data from the BELARUS TB portal program dataset, NLM, and RSNA and includes 3500 normal data points and 700 TB data points [7], and the TB dataset from Shenzen, China, comprises 662 data points [22]. Furthermore, the COVID-19 dataset, also obtained from Kaggle, is a combination of data from the PadChest dataset, a German medical school, SIRM, and GitHub, totaling 3500 COVID-19 data points [23][24].

A comparison between these two datasets is presented in Table 1. According to the table, the two types of data exhibited significant differences, especially in the COVID-19 and TB image data. The data from RSUP Dr. Rivai Abdullah tended to be cleaner and clearer, while the data obtained from Kaggle were less clean and less clear, and they contained various biases.

Table 1. Comparison Between the Chest X-Ray Image Conditions of the RSUP Dr. Rivai Abdullah And Kaggle Datasets

	RSUP	**Kaggle**
COVID-19		
TB		
Normal		

These two datasets were then grouped into three training datasets: a Kaggle dataset, an RSUP Dr. Rivai Abdullah dataset, and a combined dataset from both sources. These data were grouped in this way to obtain the best model for effectively generalizing test data.

The data were then balanced using the undersampling method for the Kaggle data and the oversampling method for the RSUP data. Undersampling involves randomly removing or eliminating some members from the data group with more instances to make it proportionate to the group with fewer instances. Conversely, oversampling is the opposite of undersampling, where the group with fewer data points is randomly increased to match the group with more data points. Oversampling was applied to the RSUP Dr. Rivai Abdullah data because the number of data points in this dataset is relatively small (in the hundreds). The aim was to mitigate the impact of significant biases when performing data duplication. On the other hand, the undersampling method was applied to the Kaggle data because it had a larger number of data points, exceeding 1000. The aim was to prevent bias resulting from image similarity due to the oversampling process, even though some data were removed. The balancing results obtained for both datasets are shown in Figure 1.

After balancing the datasets derived from both sources, i.e., RSUP Dr. Rivai Abdullah and Kaggle, the next step was data splitting. The data were divided into training, validation, and testing data. The testing data consisted of 10 images from each class (normal, TB, and COVID-19), resulting in a total of 30 images used as test data. These test data were obtained from both the RSUP Dr. Rivai Abdullah and Kaggle datasets. However, the primary test data used were from the RSUP Dr. Rivai Abdullah dataset because the main focus of the research was the ability to identify TB in X-ray images of Indonesian individuals. The data from Kaggle were only used for comparison purposes.

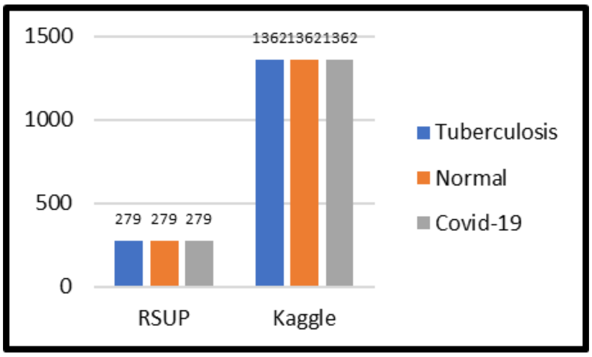


Figure 1. Result of Dataset Balancing

Once the test data were obtained, the remaining data were divided into training and validation data. The training data were used to train or build a CNN model, while the validation data were used to optimize the model obtained from the training process. The CNN model was trained using the training data, and its performance during training was evaluated using the validation data.

During the CNN model training process, the splitting ratio between the training and validation data was 80:20, meaning that 80% of the total data was used as training data, while the remaining 20% was used as validation data. The results of the data-splitting step are shown in Figure 2.

The data that had been previously divided needed to undergo a preprocessing stage before being used as inputs for the CNN model training process. The data preprocessing steps carried out in this research included cropping, grayscale conversion, resizing, scaling, and augmentation.

The cropping process was performed on some images that had backgrounds other than chest X-ray images. The cropping process was performed manually for certain images. One of the results of this cropping procedure is shown in Figure 3.

After the cropping process was completed, the next preprocessing step was to convert the images in the dataset into grayscale to ensure that each image became a grayscale image with consistent grayscale intensity levels, as some images may have had varying intensity levels. This method was employed to minimize research bias. Examples of X-ray images with different pixel intensity values are shown in Figure 4 (a). Many of these images were primarily sourced from the Kaggle dataset. The next preprocessing step was the resizing process, which was implemented to ensure that all utilized images were the same size.

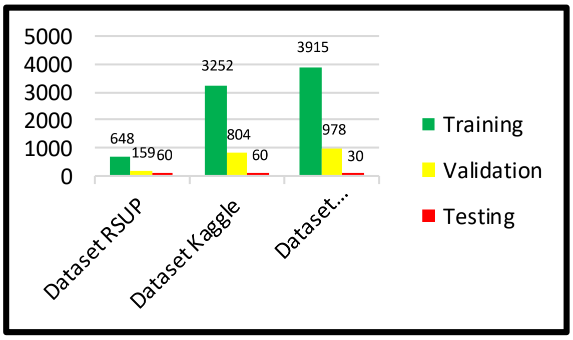


Figure 2. Splits of the Training, Validation, and Testing Data

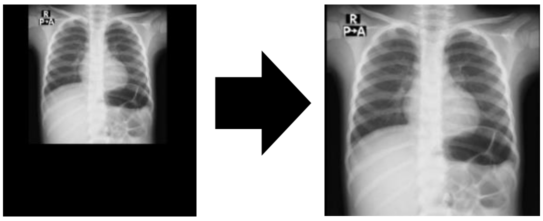


Figure 3. Sample of the Cropping Results

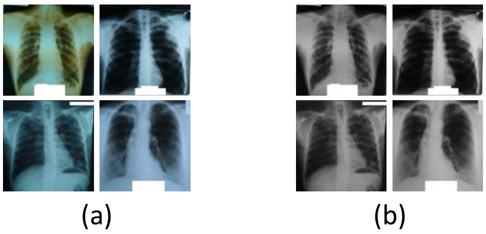


Figure 4. (a) An example of a chest X-ray image with non-gray-scale image intensity. (b) A chest X-ray image after the grayscaling process

This step was achieved by changing the Kaggle data, which originally had a size of 512 x 512 pixels and the RSUP Dr. Rivai Abdullah data, which was originally approximately 2000 x 2000 pixels, to sizes of 150 x 150 pixels. Resizing was performed to reduce the data size, as this parameter affects the computation time during the training process.

The data that were resized were then preprocessed in the scaling stage, where the images were multiplied by a value of 1/255. Images are represented as 3D arrays or tensors with values ranging from 0 to 255. Therefore, multiplying each value by 1/255 results in a value range between 0 and 1. Scaling was implemented to facilitate the neural network learning process.

The final preprocessing step was augmentation. The purpose of data augmentation is to create numerous image variations. The data augmentation strategy used included parameters such as width_shift_range=0.1, height_shift_range=0.1, zoom_range=0.1, horizontal_flip=True, rotation_range=0.1, and fill_mode='nearest'.

In this study, the training process was conducted using three CNN architectures—the AlexNet architecture (Model A), the LeNet architecture (Model B), and the modified architecture (Model C)—using the three datasets discussed above. AlexNet [25][26] and LeNet [27] were chosen because have yielded good medical image identification results. Model C is a modified architecture resulting from self-modification and development, with fewer parameters than those in AlexNet but deeper layers than those of LeNet, as shown in Table 2.

Table 2. Modified Architecture (Model C)

Layer (Type)	Output Shape	Chest X-Ray Parameter
conv2d (Conv2D)	(None, 148, 148, 32)	896
conv2d_1 (Conv2D)	(None, 146, 146, 64)	18496
max_pooling2d (MaxPooling2D)	(None, 73, 73, 64)	0
dropout (Ddrpout)	(None, 73, 73, 64)	0
conv2d_2 (Conv2D)	(None, 71, 71, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 35, 35, 64)	0
dropout_1 (Ddrpout)	(None, 35, 35, 64)	0
conv2d_3 (Conv2D)	(None, 33, 33, 128)	73856
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 128)	0
dropout_2 (Ddrpout)	(None, 16, 16, 128)	0
flatten (Flatten)	(None, 32768)	0
dense (Dense)	(None, 64)	2097216
dropout_3 (Ddrpout)	(None, 64)	0
dense_1 (Dense)	(None, 3)	195

Table 3. Training Parameters of the CNNs

Parameter	Value
Optimizer	Adam
Learning Rate	5e-5–0
Batch Size	32
Loss	Categorical Cross-Entropy
Number of Epochs	100

This architecture is used to compare the capabilities of AlexNet and LeNet in identifying conditions on chest X-rays.

The training results of these three architectures served as comparison data to determine the best CNN model, which was used in the model deployment phase in the UI. The parameters used to train the three models were the same to ensure an equitable comparison among the three architectures. The parameters used in the training process can be found in Table 3.

RESULTS AND DISCUSSION

Training Results Obtained Using Three Models

The training was conducted on Model A, Model B, and Model C using the same parameters, namely, the adaptive moment estimation (Adam) optimizer, a batch size of 32, an initial learning rate of 5×10^(-5) (which then gradually decreased until it reached a value of 0), the categorical cross-entropy loss as the loss function, and a total of 100 epochs for training. The learning rate was gradually decreased by implementing a learning rate scheduler to prevent overfitting, which is an undesired behavior in ML by which a model provides accurate predictions for the training data but not for new data. The extent of overfitting could be observed from the difference between the training accuracy and validation accuracy or that between the training loss and validation loss.

Training Using the Kaggle Dataset

The results of training and the comparison among the three models using the Kaggle dataset are shown in Figure 5. This figure shows the comparison graphs of the training and validation processes for the three models. In general, all three models experienced optimal increases in accuracy and decreases in their losses, as shown in Figures 5 (a) and (c).

Model A exhibited greater accuracy increases and lower loss decreases for both the training data and the validation data than did the other two models. This result indicates that Model A was the best-performing model in this training scenario, followed by Model C and Model B. These results show that Model C (the modified model) performed well even though it has fewer parameters than AlexNet (Model A). Additionally, the performance was LeNet (Model B) was lower than that of the modified model.

Training Using the RSUP Dataset

The training results and the comparison among the three models using the RSUP dataset are shown in Figure 6. The images provide a clearer comparison of the three models. These images show the comparison graphs produced for the training and validation processes of the three models.

In general, all three models experienced significant increases in accuracy and significant decreases in their losses.

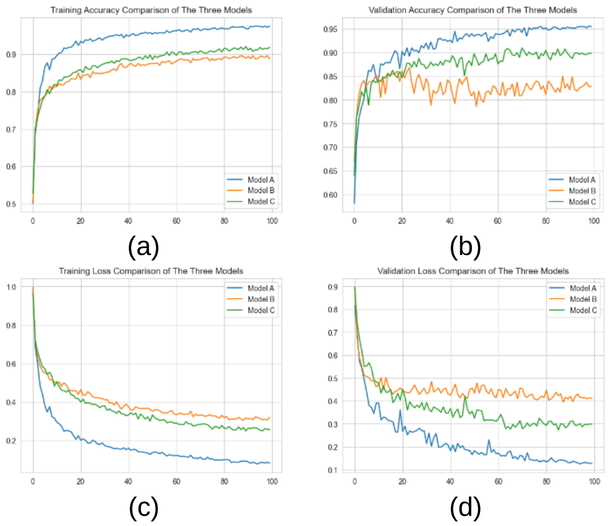


Figure 5. Comparison Graphs for the (a) Training Accuracies, (b) Validation Accuracies, (c) Training Losses, and (d) Validation Losses Achieved on the Kaggle Dataset

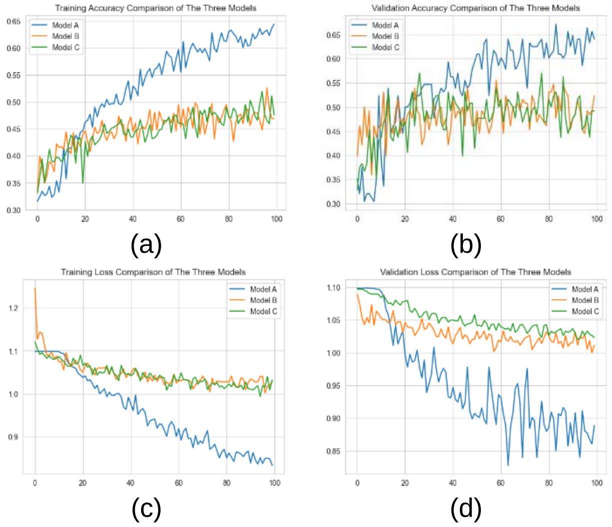


Figure 6. Comparison Graphs for the (a) Training Accuracies, (b) Validation Accuracies, (c) Training Losses, and (d) Validation Losses Achieved by the Three Models on the RSUP Dataset

Model A exhibited greater accuracy increases and greater loss decreases than did the other two models. These results indicate that Model A performed best in this training scenario, followed by Model B and Model C, with nearly identical graph comparisons. These results indicate that Model C (the modified model) demonstrated fairly good performance despite having fewer parameters than AlexNet (Model A). Additionally, the performance of LeNet (Model B) was close to that of Model C during the training process using the RSUP Dr. Rivai Abdullah data.

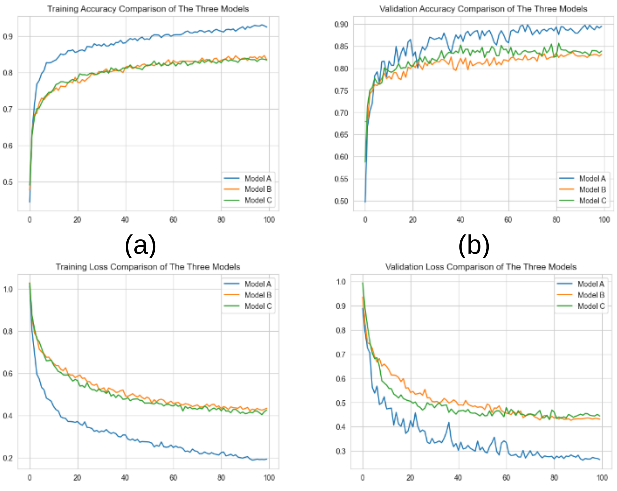
Training Using the Combined Dataset

The results of the training process and the comparison among the three models conducted on the combined dataset are shown in Figure 7. This image shows graphs of the training and validation processes of the three models. From the graphs, it can be observed that all three models experienced significant accuracy increases and loss decreases; Model A had the highest accuracy increase and the lowest loss reduction among the three, indicating that Model A was the best-performing model in this training scenario. This result was followed by the results of Models C and B, which had similar graphs. These results indicate that Model C (the modified model) performed well despite having fewer parameters than AlexNet (Model A). In contrast, LeNet (Model B) exhibited lower performance than the modified model.

The comparisons among the training and validation accuracy values produced by the three models on each dataset are shown in Table 4.

Table 5 displays the training and validation loss values yielded by the three models on each dataset.

Based on the data in Table 4 and Table 5, the final accuracy and loss values produced for both the training and validation data indicate that, overall, Model A performed best among the three models on the Kaggle data, the RSUP Dr. Rivai Abdullah data, and the combined data. Model A yielded high accuracy values and low loss values in each training process when using the three different datasets. The accuracies achieved by Model A were 97.52% on the training data and 95.50% on the validation data, with a training loss of 8.23% and a validation loss of 12.82% for the Kaggle dataset. On the RSUP Dr. Rivai Abdullah dataset, Model A achieved a training accuracy of 64.45% and a validation accuracy of 64.06%, with a training loss of 83.32% and a validation loss of 88.88%. On the combined dataset, Model A achieved a training accuracy of 92.43% and a validation accuracy of 89.48%, with a training loss of 19.18% and a validation loss of 26.44%. On the other hand, Model C, which is a modified and self-developed model, achieved better performance than Model B in terms of the accuracy values produced during each training process using the three datasets. Regarding the loss, Model B performed slightly better when training with the RSUP Dr. Rivai Abdullah dataset, but its results were not significantly different from those of Model C. Thus, this shows that LeNet, which has fewer layers, is not sufficient for obtaining a good model for the chest X-ray dataset.



(a) (b)

(c) (d)

Figure 7. Comparison graphs produced by the three models on the combined dataset, showing their (a) training accuracies, (b) validation accuracies, (c) training losses, and (d) validation losses

Table 4. Training and Validation Accuracies of the Three Models (%)

Model	Kaggle		RSUP		Combined	
	TA	**VA**	**TA**	**VA**	**TA**	**VA**
A	97.5	95.5	64.4	64.1	92.4	89.5
B	88.7	82.9	46.9	52.3	83.4	83.1
C	91.9	89.9	47.6	49.2	83.4	83.8

Note: TA = Training Accuracy, VA = Validation Accuracy

Table 5. Training and Validation Losses of the Three Models (%)

Model	Kaggle		RSUP		Combined	
	TL	**VL**	**TL**	**VL**	**TL**	**VL**
A	8.23	12.8	83.3	88.9	19.2	26.4
B	32.1	41.1	103	101	43.3	43.1
C	25.7	29.9	103.	102	42.5	44.4

Note: TA = Training Accuracy, VA = Validation Accuracy

CNN Model Testing Results

Testing the Trained Models with the Kaggle Dataset

The testing procedure was conducted using previously prepared testing data to evaluate the performance of each model.

Table 6 and Table 7 show that all three models trained using the Kaggle dataset performed well in identifying testing data from Kaggle. However, the models performed poorly when used to identify testing data from the RSUP Dr. Rivai Abdullah dataset. These results may have occurred because the training was conducted. Using only Kaggle data, which has different characteristics compared to those of the RSUP Dr. Rivai Abdullah data.

Model A achieved an accuracy of up to 90% on the testing data using the Kaggle dataset. According to the confusion matrix, Model A correctly identified ten chest X-ray images of TB, which is the primary focus of this research.

Table 6. Confusion Matrices Produced by Testing the CNN Models Trained on the Kaggle Dataset

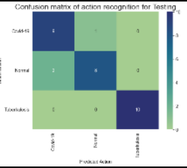
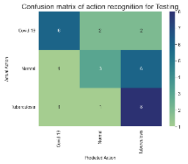
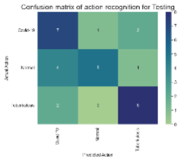
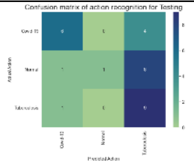
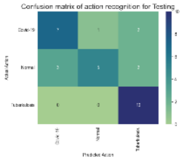
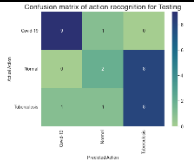
	Kaggle Dataset	**RSUP Dataset**
Model A (AlexNet)		
Model B (LeNet)		
Model C (Modified)		

Table 7. Performance Metrics Yielded by the Three CNN Models Trained Using the Kaggle Dataset

			Model A	Model B	Model C
Kaggle Dataset	Accuracy		0.9	0.66	0.73
	Sensitivity	COVID-19	0.9	0.7	0.7
		Normal	0.8	0.5	0.5
		TB	1.0	0.8	1.0
	Specificity	COVID-19	0.9	0.7	0.85
		Normal	0.95	0.95	0.95
		Tuberculosis	1.0	0.85	0.8
	Precision	COVID-19	0.82	0.54	0.7
		Normal	0.89	0.83	0.83
		Tuberculosis	1.0	0.73	0.71
	F1 Score	COVID-19	0.86	0.61	0.7
		Normal	0.84	0.62	0.62
		Tuberculosis	1.0	0.76	0.83
RSUP Dataset	Accuracy		0.57	0.53	0.63
	Sensitivity	COVID-19	0.6	0.6	0.9
		Normal	0.3	0.1	0.2
		Tuberculosis	0.8	0.9	0.8
	Specificity	COVID-19	0.9	0.9	0.95
		Normal	0.85	1.0	0.9
		Tuberculosis	0.6	0.4	0.6
	Precision	COVID-19	0.75	0.75	0.9
		Normal	0.5	1.0	0.5
		Tuberculosis	0.5	0.43	0.5
	F1 Score	COVID-19	0.67	0.67	0.9
		Normal	0.37	0.18	0.28
		Tuberculosis	0.61	0.58	0.61

Meanwhile, for testing using the RSUP Dr. Rivai Abdullah dataset, Model C performed better, with an accuracy rate of 63%. However, based on the confusion matrix, Model B could better identify ten chest X-ray images of TB correctly.

Testing the Trained Models with the RSUP Dataset

Table 8 and Table 9 show that the three models trained using the RSUP dr. Rivai Abdullah dataset could only identify images in the testing data originating from this dataset. All three models exhibited poor performance, with accuracies below 50%. This result indicates that all three models suffer from underfitting due to the limited amount of available data. Regarding the performance of the three models, Model A performed best during testing on both the Kaggle dataset and the RSUP dataset, achieving accuracy rates of 30% on the Kaggle testing data and 43% on the RSUP Dr. Rivai Abdullah testing data. Model C, on the other hand, achieved accuracy rates of 26% on the Kaggle testing data and 40% on the RSUP Ddr. Rivai Abdullah testing data, while Model B performed worst, with accuracy rates of 23% on the Kaggle testing data and 36% on the RSUP Dr. Rivai Abdullah testing data

Table 8. Confusion Matrices Produced by Testing the CNN Models Trained on the RSUP Dataset

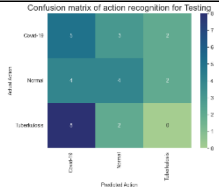
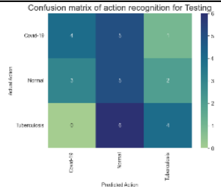
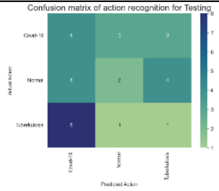
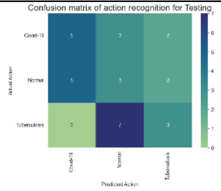
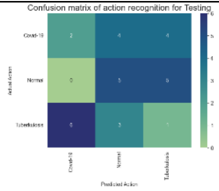
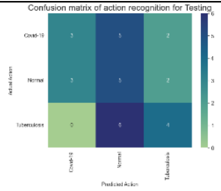
	Kaggle Dataset	RSUP Dataset
Model A (AlexNet)		
Model B (LeNet)		
Model C (Modified)		

Table 9. Performance Metrics Yielded by the Three CNN Models Trained Using the RSUP Dataset

			Model A	Model B	Model C
Kaggle Dataset	Accuracy		0.3	0.23	0.26
	Sensitivity	COVID-19	0.5	0.4	0.2
		Normal	0.4	0.2	0.5
		Tuberculosis	0.0	0.1	0.1
	Specificity	COVID-19	0.4	0.4	0.7
		Normal	0.75	0.8	0.65
		Tuberculosis	0.8	0.65	0.55
	Precision	COVID-19	0.29	0.25	0.25
		Normal	0.44	0.33	0.42
		Tuberculosis	0.0	0.12	0.1
	F1 Score	COVID-19	0.37	0.31	0.22
		Normal	0.42	0.25	0.45
		Tuberculosis	0.0	0.11	0.1
RSUP Dataset	Accuracy		0.43	0.36	0.4
	Sensitivity	COVID-19	0.4	0.5	0.3
		Normal	0.5	0.3	0.5
		Tuberculosis	0.4	0.3	0.4
	Specificity	COVID-19	0.85	0.75	0.85
		Normal	0.45	0.5	0.45
		Tuberculosis	0.85	0.8	0.8
	Precision	COVID-19	0.57	0.5	0.5
		Normal	0.31	0.23	0.31
		Tuberculosis	0.57	0.43	0.5
	F1 Score	COVID-19	0.47	0.5	0.37
		Normal	0.38	0.26	0.38
		Tuberculosis	0.47	0.35	0.44

Testing the Trained Models with the Combined Dataset

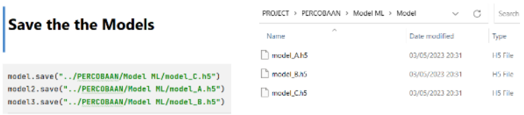
Table 10 and Table 11 show that Model A trained with the combined dataset achieved a high level of testing accuracy using both the Kaggle and RSUP Dr. Rivai Abdullah datasets. With the RSUP Dr. Rivai Abdullah dataset, Model A achieved an accuracy of 88% and could correctly detect 7 out of 10 X-ray images in the COVID-19 class, 10 out of 10 images in the normal class, and 7 out of 10 images in the TB class. Model B achieved an accuracy of 53%, correctly detecting 3 out of 10 images in the COVID-19 class, 7 out of 10 images in the normal class, and 6 out of 10 images in the TB class. Model C also achieved an accuracy rate of 53%, correctly detecting 6 out of 10 images in the COVID-19 class, 3 out of 10 images in the normal class, and 7 out of 10 images in the TB class.

From the testing performed using the three different trained models—Model A, Model B (LeNet), and Model C (modified architecture)—as shown in Table 7, Table 9, and Table 11, it can be observed that Model A, which is AlexNet, was more accurate than Model B and Model C. This indicates that the AlexNet model is suitable for deployment to TB using chest X-rays at the local hospital, RSUP Dr. Rivai Abdullah. The increased number of layers and parameters may influence the model's learning capability. However, Model C, the modified architecture, also has the potential to be used instead of LeNet, despite having fewer parameters.

Steps to Deploy the Model to a UI

At the end of the CNN model training process, the model is saved in H5 (HDF5) file format, as shown in Figure 8, to facilitate easier model deployment using the Python programming language. H5-formatted models can store large amounts of numeric data that can be easily manipulated using NumPy. H5-formatted models are widely used because they have the ability to store thousands of datasets in a single file.

After the CNN model is created, the selected model is uploaded to the UI environment within the Hugging Face platform, which is an online platform for website hosting. All the files within the environment are shown in Figure 9.



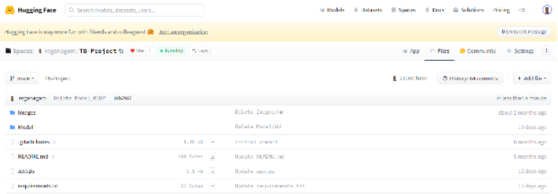
Save the the Models

```
model.save("../PERCOBAAN/Model ML/model_C.h5")
model2.save("../PERCOBAAN/Model ML/model_A.h5")
model3.save("../PERCOBAAN/Model ML/model_B.h5")
```

PROJECT › PERCOBAAN › Model ML › Model

Name	Date modified	Type
model_A.h5	03/05/2023 20:31	H5 File
model_B.h5	03/05/2023 20:31	H5 File
model_C.h5	03/05/2023 20:31	H5 File

Figure 8. Process of Saving the CNN Model to an HDF5 File



reganagam		Delete Model_RSUP 16b7987	in less than a minute
Images		Delete Images/nm	about 2 months ago
Model		Delete Model/old	19 days ago
.gitattributes	1.48 kB	initial commit	6 months ago
README.md	249 Bytes	Update README.md	5 months ago
app.py	1.1 kB	Update app.py	13 days ago
requirements.txt	33 Bytes	Update requirements.txt	13 days ago

Figure 9. Files in the Hugging Face Environment.

Table 10. Confusion Matrix Produced by Testing the CNN Models Trained on the Combined Dataset

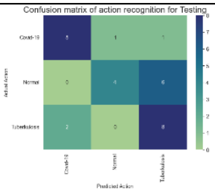
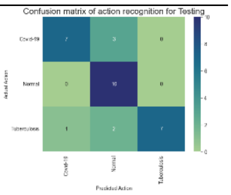
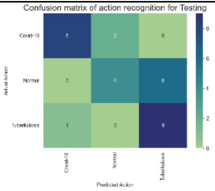
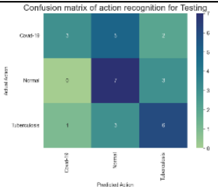
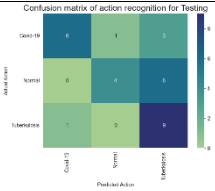
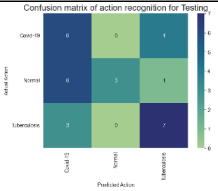
	Dataset Kaggle	Dataset RSUP
Model A (AlexN)		
Model B (LeNet)		
Model C (Modified)		

Table 11. Performance Evaluation Results Yielded by the Three CNN Models Trained Using the Combined Dataset

			Model A	Model B	Model C
Kaggle Dataset	Accuracy		0.67	0.7	0.63
	Sensitivity	COVID-19	0.8	0.8	0.6
		Normal	0.4	0.4	0.4
		Tuberculosis	0.8	0.9	0.9
	Specificity	COVID-19	0.9	0.95	0.95
		Normal	0.95	0.9	0.95
		Tuberculosis	0.65	0.7	0.55
	Precision	COVID-19	0.8	0.89	0.86
		Normal	0.8	0.67	0.8
		Tuberculosis	0.53	0.6	0.5
	F1 Score	COVID-19	0.8	0.84	0.71
		Normal	0.53	0.5	0.53
		Tuberculosis	0.64	0.72	0.64
RSUP Dataset	Accuracy		0.88	0.53	0.53
	Sensitivity	COVID-19	0.7	0.3	0.6
		Normal	1.0	0.7	0.3
		Tuberculosis	0.7	0.6	0.7
	Specificity	COVID-19	0.95	0.95	0.55
		Normal	0.75	0.6	1.0
		Tuberculosis	1.0	0.75	0.75
	Precision	COVID-19	0.87	0.75	0.4
		Normal	0.67	0.47	1.0
		Tuberculosis	1.0	0.54	0.58
	F1 Score	COVID-19	0.78	0.43	0.48
		Normal	0.8	0.56	0.46
		Tuberculosis	0.82	0.57	0.63

The following components are found in the environment: the "Images" folder, which contains several chest X-ray images used as samples in the UI; the "Model" folder, which contains the selected CNN model file in the H5 format; the "gitattributes" file, which is a special attribute file for the GIT-distributed version control system or a path giver for each component; the "README.md" file, which contains metadata about the components used in the environment; the "requirements.txt" file, which includes the libraries used in the model deployment process (TensorFlow, Keras, and Gradio); and the "app.py" file, which is a Python file containing the Python code for deploying the CNN model.

Graphic UI

The final appearance of the UI is shown in Figure 10. This UI can be accessed through the following URL: "https://reganagam-tb-project.hf.space". The UI includes a title indicating its purpose, which is "Tuberculosis (TB) Detection Using Chest X-Rays", an input column for images, an output column for the produced detection results, some sample images that can be used, a "submit" button, and a "clear" button for removing the output if one wishes to perform further image detection tasks.

The disease detection process with the created UI begins by accessing the provided website page. Then, the input, which is a chest X-ray image in the jpg/png format, can be added to the UI by either dragging the image from one's local computer to the UI input section (dropping) or clicking on the UI input section directly. Selecting the image for detection from the local computer. The image input process is shown in Figure 11.

After the desired image is added to the UI, the next step is to press the "submit" button to initiate the image detection process or press the "clear" button to remove the image if the wrong image is added. The input image can be of any size because the program in the UI can directly resize the image to 150 × 150 pixels, as specified by the input size set during the CNN model training procedure. Some samples in the UI can also be used for testing the UI by clicking on one of the images. Then, the detection result produced for the input image appears immediately in the output column.

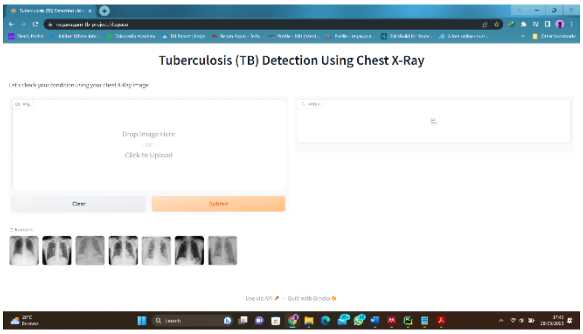

Tuberculosis (TB) Detection Using Chest X-Ray
Let's check your condition using your chest X-Ray image!
Drop Image Here
- or -
Click to Upload
Clear
Submit
Use via API · Built with Gradio

Figure 10. Final Display of the UI.

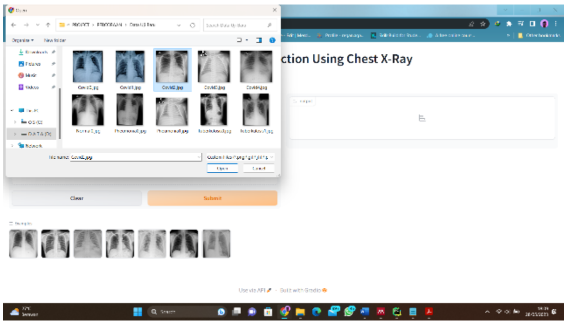


Figure 11. Chest X-ray Image Inputting Process of the UI.

The output displayed in Figure 12 shows the condition of the chest X-ray image (COVID-19, TB, or normal), with the lower part indicating the UI's confidence (as a percentage) in the prediction result.

Usability Testing with a Healthcare Professional

This research aimed to develop a UI that can be used to assist professionals in identifying TB, making usability testing with healthcare professionals crucial. In the usability testing stage, a healthcare professional, specifically a radiologist from RSUP, Dr. Rivai Abdullah, who is an expert in diagnosing diseases through medical images, was involved. The radiologist was asked to use the UI that had been built.

The testing was also conducted using 10 new samples prepared by RSUP Dr. Rivai Abdullah. Based on the testing results shown in Table 12, the UI built using the AlexNet model could correctly detect 7 out of the 10 test images, achieving a success rate of 70%. During testing, the average speed of the UI when detecting all ten images did not exceed 15 seconds. Therefore, the developed UI is user-friendly. The testing results revealed that the detection speed of the UI depends on the input image size.

After the radiology specialist conducted direct tests on the created UIs, they were also asked to complete a questionnaire.

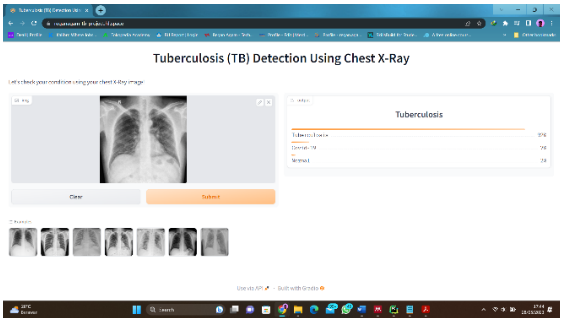

Tuberculosis (TB) Detection Using Chest X-Ray
Let's check your condition using your chest X-Ray image!
Tuberculosis
Tuberculosis
Covid-19
Normal
Clear
Submit
Use via API · Built with Gradio

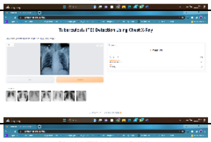
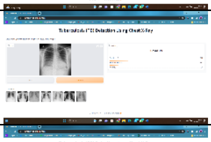
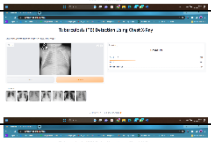
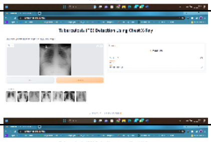
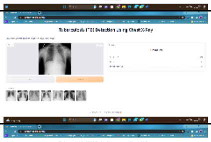
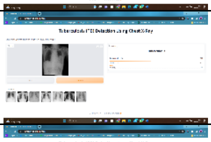
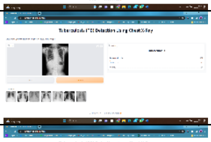
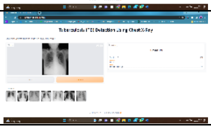
Figure 12. Output of the UI.

This questionnaire aimed to gather experts' perspectives regarding the use of the developed website for TB identification purposes. Overall, the questionnaire results indicated that the UI is easy to use, has a pleasing appearance, and is helpful for healthcare professionals when performing initial screenings or diagnoses However, doctors still need to make a final disease diagnosis.

During the process of collecting data for the questionnaire, the doctor was also interviewed to provide explanations regarding the TB classes for severity level classification, and further development should be pursued in the future. Based on the interviews with the radiology specialist, it was stated that, in general, TB has two severity levels: nonsevere and severe. However, obtaining chest X-ray data for nonsevere patients is very difficult because patients only undergo chest X-ray imaging when the disease reaches a severe stage.

Patients with nonsevere TB do not exhibit any symptoms before progressing to a severe stage. Furthermore, many people are infected with TB, which can become severe if their immune system is weakened. TB is still a highly contagious and uncontrolled epidemic, even on a global scale.

Table 12. UI testing using the new dataset

Testing Data	Detection	Actual (Medical Reading)	Prediction (UI)
		COVID-19	COVID-19
		COVID-19	COVID-19
		COVID-19	COVID-19
		COVID-19	COVID-19
		COVID-19	COVID-19
		Normal	COVID-19
		Pneumonia/ COVID-19	Tuberculosis
		Pneumonia/ COVID-19	COVID-19
		Tuberculosis	Tuberculosis
		Tuberculosis	COVID-19

CONCLUSION

Based on the conducted research, CNNs can be implemented to interpret X-ray images for detecting tuberculosis (TB). AlexNet (Model A) outperformed LeNet (Model B) and the custom-developed model (Model C). This result was observed from the model accuracy values produced during training and testing. Model A achieved training and validation accuracy of 97.52% and 95.50%, respectively, with training and validation losses of 8.23% and 12.82%, when trained on the Kaggle dataset. When trained on the RSUP Dr. Rivai Abdullah dataset, Model A achieved a training accuracy of 64.45%, a validation accuracy of 64.06%, a training loss of 83.32%, and a validation loss of 88.88%. On the combined dataset, this model achieved training and validation accuracies of 92.43% and 89.48%, respectively, with training and validation losses of 19.18% and 26.44%. However, the training process of Model A is significantly longer, and its size is much larger than those of Models B and C due to the significantly greater number of layers used. The model trained with the AlexNet architecture (Model A) was subsequently used in the deployment of a UI for detecting TB across three classes. The web-based UI utilizing the Gradio library demonstrated good performance, achieving a detection accuracy of 70% for new data from RSUP Dr. Rivai Abdullah. This UI can be used for the early-stage diagnosis (screening) of TB patients.

ACKNOWLEDGMENT

The authors would like to thank the radiology specialists at Dr. Rivai Abdullah Hospital for their help in interpreting the X-ray images and during the implementation of the X-ray data.

REFERENCES

[1] S.S. Alsayed and H. Gunosewoyo, “Tuberculosis: Pathogenesis, Current Treatment Regimens and New Drug Targets,” *International Journal of Molecular Sciences,* vol. 24, no. 6, 2023, doi: 10.3390/ijms24065202

[2] World Health Organization, “Global Tuberculosis Report 2022,” *World Health Organization*, Geneva, 2022. [Online]. Available: https://www.who.int/publications/i/item/9789240061729

[3] O. Yadav, K. Passi, and C. K. Jain, “Using Deep Learning to Classify X-ray Images of Potential Tuberculosis Patients,” *Proc. - 2018 IEEE Int. Conf. Bioinforma. Biomed. BIBM 2018*, no. September 2019, pp. 2368–2375, 2019, doi: 10.1109/BIBM.2018.8621525.

[4] D. Verma, C. Bose, N. Tufchi, K. Pant, V. Tripathi, and A. Thapliyal, “An efficient framework for identification of Tuberculosis and Pneumonia in chest X-ray images using Neural Network,” *Procedia Computer Science.*, vol. 171, no. 2019, pp. 217–224, 2020, doi: 10.1016/j.procs.2020.04.023.

[5] S. Stirenko *et al.*, “Chest X-Ray Analysis of Tuberculosis by Deep Learning with Segmentation and Augmentation,” *2018 IEEE 38th Int. Conf. Electron. Nanotechnology, ELNANO 2018 - Proc.*, pp. 422–428, 2018, doi: 10.1109/ELNANO.2018.8477564.

[6] S. Aulia and S. Hadiyoso, “Tuberculosis Detection in X-Ray Image Using Deep Learning Approach with VGG-16 Architecture,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI Journal).*, vol. 8, no. 2, p. 290, 2022, doi: 10.26555/jiteki.v8i2.23994.

[7] T. Rahman *et al.*, “Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization,” *IEEE Access*, vol. 8, pp. 191586–191601, 2020, doi: 10.1109/ACCESS.2020.3031384.

[8] K. Munadi, K. Muchtar, N. Maulina, and B. Pradhan, “Image Enhancement for Tuberculosis Detection Using Deep Learning,” *IEEE Access*, vol. 8, pp. 217897–217907, 2020, doi: 10.1109/ACCESS.2020.3041867.

[9] N. Fitriyah, L. D. Kurniawati, E. Purwanti, and S. D. Astuti, “Pulmonary Disease Pattern Recognition on X-Ray Radiography Image Using Artificial Neural Network (ANN) Method,” *Journal of Physics: Conference Series (JPCS)*, vol. 1505, no. 1, 2020, doi: 10.1088/1742-6596/1505/1/012065.

[10] A. Achmad, and A. D. Achmad, “Backpropagation Performance Against Support Vector Machine in Detecting Tuberculosis Based on Lung X-Ray Image,” *Proceedings of the First International Conference on Materials Engineering and Management - Engineering Section (ICMEMe 2018),* vol. 165, 2018, pp. 84–88, doi: 10.2991/icmeme-18.2019.19.

[11] Muhathir, T. T. S. Sibarani and Al-Khowarizmi, “Analysis K-Nearest Neighbors (KNN) in Identifying Tuberculosis Disease (Tb) By Utilizing Hog Feature Extraction,” *International Journal of Computer Science and Information Technology,* vol. 1, no. 1, pp. 33–38, 2020.

[12] O. Hrizi *et al.*, “Tuberculosis Disease

Diagnosis Based on an Optimized Machine Learning Model," *J. Healthc. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/8950243.

[13] Z. Ul Abideen *et al.*, "Uncertainty assisted robust tuberculosis identification with bayesian convolutional neural networks," *IEEE Access*, vol. 8, pp. 22812–22825, 2020, doi: 10.1109/ACCESS.2020.2970023.

[14] M. Owais, M. Arsalan, T. Mahmood, Y. H. Kim, and K. R. Park, "Comprehensive computer-aided decision support framework to diagnose tuberculosis from chest X-Ray images: Data mining study," *JMIR Med. Informatics*, vol. 8, no. 12, pp. 1–24, 2020, doi: 10.2196/21790.

[15] S. M. Fati, E. M. Senan, and N. ElHakim, "Deep and Hybrid Learning Technique for Early Detection of Tuberculosis Based on X-ray Images Using Feature Fusion," *Appl. Sci.*, vol. 12, no. 14, 2022, doi: 10.3390/app12147092.

[16] Q. H. Nguyen *et al.*, "Deep Learning Models for Tuberculosis Detection from Chest X-ray Images," *2019 26th Int. Conf. Telecommun. ICT 2019*, pp. 381–385, 2019, doi: 10.1109/ICT.2019.8798798.

[17] N. F. Aurna, M. A. Yousuf, K. A. Taher, A. K. M. Azad, and M. A. Moni, "A classification of MRI brain tumor based on two stage feature level ensemble of deep CNN models," *Computers in Biology and Medicine*, vol. 146, no. April, p. 105539, 2022, doi: 10.1016/j.compbiomed.2022.105539.

[18] M. Rahman, Y. Cao, X. Sun, B. Li, and Y. Hao, "Deep pre-trained networks as a feature extractor with XGBoost to detect tuberculosis from chest X-ray," *Computers & Electrical Engineering*, vol. 93, no. April, p. 107252, 2021, doi: 10.1016/j.compeleceng. 2021.107252.

[19] B. Herman, W. Sirichokchatchawan, S. Pongpanich, and C. Nantasenamat, "Development and performance of CUHASROBUST application for pulmonary rifampicin-resistance tuberculosis screening in Indonesia," *PLoS One*, vol. 16, no. 3 March, pp. 1–19, 2021, doi: 10.1371/journal.pone.0249243.

[20] M. H. Lino Ferreira da Silva Barros *et al.*, "Benchmarking machine learning models to assist in the prognosis of tuberculosis," *Informatics*, vol. 8, no. 2, pp. 1–17, 2021, doi: 10.3390/informatics8020027.

[21] A. Siddique, K. Shuakat, and T. Jan, "An Intelligent Mechanism to Detect Multi-Factor Skin Cancer," *Diagnostics,* vol. 14, no. 13, 2024, doi: 10.3390/diagnostics14131359

[22] D. Barušauskas, "Tuberculosis Chest X-rays (Shenzhen)," *Kaggle*, 2020. [Online]. Available: https://www.kaggle.com/datasets/raddar/tuberculosis-chest-xrays-shenzhen

[23] M. E. H. Chowdhury *et al.*, "Can AI Help in Screening Viral and COVID-19 Pneumonia?," *IEEE Access*, vol. 8, pp. 132665–132676, 2020, doi: 10.1109/ACCESS.2020.3010287.

[24] [L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, no. 1., pp. 1-74, 2021. doi: 10.1186/s40537-021-00444-8.

[25] R. Kurniawan et al., "Classification of palm oil fruit ripeness based on AlexNet deep Convolutional Neural Network," *SINERGI*, vol. 29, no. 1, pp. 241-250, 2025, doi: 10.22441/sinergi.2025.1.

[26] M. Nijiati, J. Ma, C. Hu, A. Tuersun, A. Abulizi, A. Kelimu, D. Zhang, G. Li, and X. Zou, "Artificial Intelligence Assisting the Early Detection of Active Pulmonary Tuberculosis from Chest X-ray: A Population-Based Sudy," *Frontiers in Molecular* Biosciences, vol. 9, 2022, doi: 10.3389/fmolb.2022.874475

[27] M. Toğaçar, B. Ergen, and Z. Cömert, "Detection of lung cancer on chest CT images using minimum redundancy maximum relevance feature selection method with convolutional neural networks," *Biocybern. Biomed. Eng.*, vol. 40, no. 1, pp. 23–39, 2020, doi: 10.1016/j.bbe.2019.11. 004.